Knauf, Rainer; Tsuruta, Setsuo; Ihara, Hirokazu;
Gonzalez, Avelino J.; Kurbad, Torsten :

Improving AI systems' dependability by utilizing historical knowledge

Zuerst erschienen in:
Proceedings / 10th IEEE Pacific Rim International Symposium on Dependable Computing, 3-5 March 2004, Papeete, Tahiti, French Polynesia. - Los Alamitos, Calif. [u.a.] : IEEE Computer Society, S. 343–352
DOI: 10.1109/PRDC.2004.1276590

Improving AI Systems' Dependability by Utilizing Historical Knowledge

Rainer Knauf
Technical University of Ilmenau
School of Computer Science and Automation
Chair of Artificial Intelligence
PO Box 10 05 65, 98684 Ilmenau, Germany
rainer.knauf@tu-ilmenau.de

Setsuo Tsuruta & Hirokazu Ihara
Tokyo Denki University
School of Information Environment
2-1200 Musai-gakuendai, Inzai
Chiba, 270–1382, Japan
tsuruta@sie.dendai.ac.jp, ihara@coral.ocn.ne.jp

Avelino J. Gonzalez
University of Central Florida
School of Electrical Engineering and Computer Science
Orlando, FL 32816-2450, USA
gonzalez@pegasus.cc.ucf.edu

Torsten Kurbad
TK-WebArt
Chaussee 72
37345 Großbodungen, Germany
torsten@tk-webart.de

Abstract

A TURING *Test is a promising way to validate AI systems which usually have no way to proof correctness. However, human experts (validators) are often too busy to participate in it and sometimes have different opinions per person as well as per validation session. To cope with these and increase the validation dependability, a Validation Knowledge Base (VKB) in Turing Test – like validation is proposed. The VKB is constructed and maintained across various validation sessions. Primary benefits are (1) decreasing validators' workload, (2) refining the methodology itself, e.g. selecting dependable validators using VKB, and (3) increasing AI systems' dependabilities through dependable validation, e.g. support to identify optimal solutions. Finally, Validation Experts Software Agents ($VESA$) are introduced to further break limitations of human validator's dependability. Each $VESA$ is a software agent corresponding to a particular human validator. This suggests the ability to systematically "construct" human–like validators by keeping personal validation knowledge per corresponding validator. This will bring a new dimension towards dependable AI systems.*

1. Introduction

Recently, intelligent systems are getting larger and more complex, making it difficult to develop and maintain such complex systems. The validation of these systems can be particularly difficult. Albeit for conventional (non AI) computer software, validation has been defined by ADRION [1] as *'... the determination of the correctness of the final program or software ... with respect to the user needs and requirements'* and later by BARR [2] as *'... a dynamic process. Determine that the system is behaving in accordance with specification. The conclusion of the system resembles that of the human expert who provided knowledge for the system.'* In the context of validation of intelligent systems, it is quite clear, that human performance is a benchmark of a system's validity [6] [1].

However, often each expert has a different opinion about the question, whether or not such a system has the correct behavior with respect to the users' needs. Sometimes, an expert makes judgments different from his previous ones, even in the same context. Thus, there are also limitations to the dependability of the validation by experts. Furthermore, experts are too busy to spend much time in system validation. Thus, the experts' workload for system validation is a serious issue.

To decrease this workload of the experts, the importance to store and use historical validation results / knowledge was discussed and a Validation Knowledge Base (VKB) was proposed in [18]. Subsequently, the idea to use such a VKB for supporting validation performed in the TURING Test – like approach [20] was proposed in [9].

In this paper, this idea is more logically and concretely discussed from the aspect of increasing validation dependability and system dependability. Additionally, further ideas of utilizing a VKB to increase the system dependability,

1 For a comprehensive discussion of definitions for the terms verification and validation and their particular meanings with respect to intelligent systems, see [6].

0-7695-2076-6/04 $20.00 © 2004 IEEE

through more complete validation are described. This aims at breaking limitations of dependability of human validators. For example, these ideas include an approach concerning the selection of dependable validation experts by using the VKB and an approach called $VESA$ (Validation Experts Software Agent). Though discussed mainly for rule–base systems first, it can be extended to other AI / intelligent systems such as case–based systems and, in the future, possibly to general complex systems.

Generally, an intelligent system's dependability corresponds to the correctness of its incorporated Knowledge Base (KB), but many of the AI systems do not have a commonly accepted knowledge standard. The only way to ensure dependability has been to adjust it with dependable human expertise. Thus, these adjustments, called **validation and refinement**, turned out to be a key issue for the practical use of such systems. In the framework [8] and [12] to conduct a five step validation and refinement process for rule–based systems, the result is highly influenced by the quality of interaction with human experts. Their excessive involvement is both time consuming and cost inefficient. In addition, human validators or experts who validate AI systems, may not always be available or even willing to run the given tasks, thereby causing delays to the entire process. In [19] this is summarized as *"the bottleneck in acquiring validation knowledge from experts who are busy."*

In fact, the framework in [12] has several drawbacks:

- The new topical domain knowledge gained by the validation process is acquired as an optimal (best rated) solution for each executed test case, but recorded rather implicitly as a restructured rule base that maps each test case to its optimal solution.
- The gained "experience" can't serve as a launch pad. Therefore, it can not be reused for sessions with other systems of the same application domain or for future sessions with a different human expertise (other expert panels, new topical insights, etc.).

Since validation is considered an on–going or repeated process and KBs themselves are the subject of validation, it might be necessary to urge experts to provide the same knowledge many times. Though intelligent systems must be continually or periodically validated to ensure correctness vis–a–vis the latest findings, it is very unlikely that major changes are expected from one validation session to the next for an AI system in a long–term practical application. This implies that the knowledge used in validation, namely the set of test cases including their best rated solutions as well as their authors, must persist from one validation exercise to the next. Thus, a way to store, manage, and maintain validation knowledge is required for any practical approach to validation. This could provide a vehicle for long–term management and improvement of the validation process for intelligent systems.

Though different system users might have different validation criteria, and thus different test case sets will be generated, the knowledge gained by the validation of a system can be reused for the validation of other copies of the system, almost equivalent systems or mostly similar systems. Applying this technique would effectively limit the workload of human validators, which makes the validation procedure more practical.

One approach to store the explicit knowledge gained in a validation process has been introduced by TSURUTA [19]. Here, the authors propose a VKB, which is basically a library of test cases used in previous validation sessions. The basic idea is to keep test cases with well–evaluated solutions along with a time stamp and the solution's author provided in a time-consuming TURING Test – like interrogation to reuse them for subsequent validation sessions. Combined with the validation framework developed by KNAUF et al. [12], the VKB clearly supports the rule base validation and refinement process.

The validation procedure, as developed so far, covers five steps: (1) test case generation, (2) test case experimentation, (3) evaluation of results, (4) validity assessment, and (5) system refinement. These steps can be performed iteratively. The most expensive part of this framework is the test case experimentation, because the test cases have to be solved and rated by both the system under examination and the humans who perform the examination.[2] Firstly, this step is intentionally supported by the VKB. Secondly, with a view towards dependability, the VKB is applied for other useful purposes, especially to break the limitation of the dependability of the validation by experts: (1) It can be used to improve the validation methodology itself (e.g. to select experts for the validation panel), (2) it might provide a good basis to develop appropriate domain–related validation criteria, and (3) it can be used to identify an optimal solution among several candidate solutions. The usage of the VKB from two points of view, (1) the one of the Knowledge Engineer and (2) the one of a validating expert is outlined in figure 1. Here, two steps of the validation methodology, namely the *test case generation* (which works in two steps producing (1) a *quasi exhaustive set of test cases* $QuEST$ and a *reasonable set of test cases* $ReST$) and the *test case experimentation* are considered with respect to the role of the VKB.

Lastly, as an extension of the VKB, the concept of $VESA$ is described shortly. The purpose of $VESA$ is to further break the limitations of human experts' dependability through keeping personal validation knowledge such as

2 In the process not only the system's solutions, but also the solutions provided by humans are examined. The latter is performed to estimate the experts' competence for each particular test case.

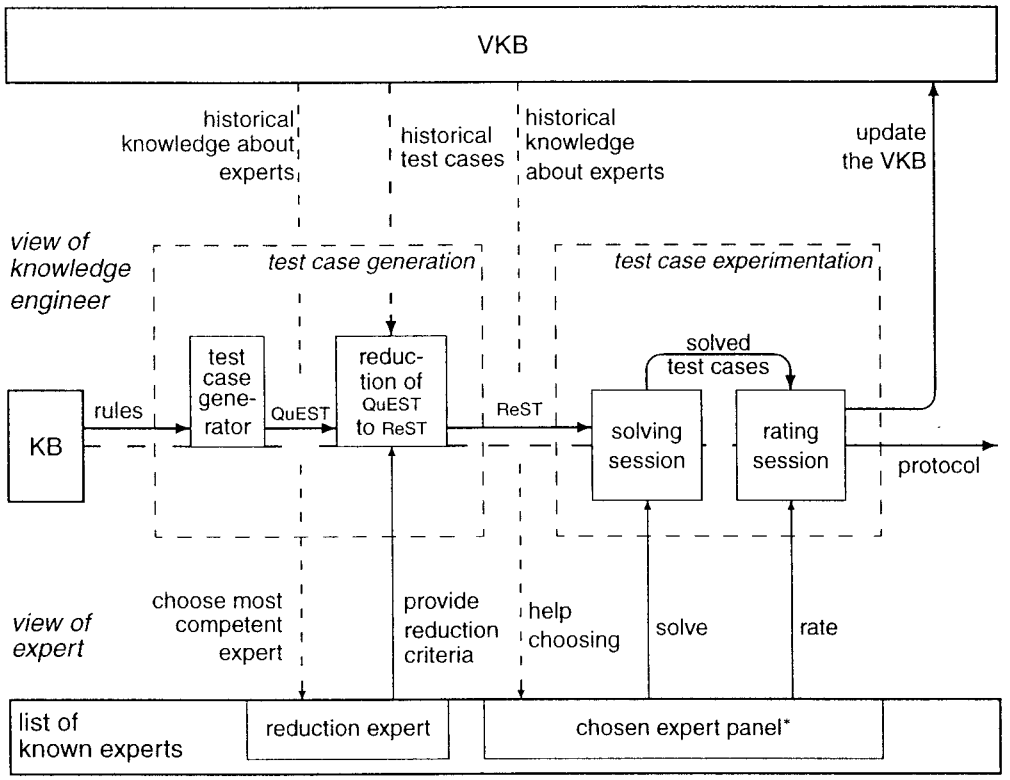

Figure 1. Involvement of the VKB in the validation process

previous validation judgments or experiences of each human expert. Each $VESA$ is an intelligent avatar corresponding to each human validation expert. This extension of the VKB will bring a new dimension for the validation and the dependability of AI systems.

After a short introduction to the validation framework of KNAUF [12] in section 2 and the original concept of the VKB by TSURUTA [19] in section 3, section 4 details the idea of utilizing TSURUTA's concept in KNAUF's framework. This is supplemented by an introduction of the software tools developed so far for this purpose.

2. The Turing Test Methodology

The validation framework introduced in [8] and [12] consists of five steps, which can be performed in cycles (see figure 2):

1. **Test case generation**: Here, an appropriate set of test cases $[TestData, ExpectedOutput]$ is generated. This set meets the competing requirements (a) *Coverage* of all possible combinations of inputs which expands the number of test cases to ensure completeness in coverage, and (b) *efficiency* which limits the number of test cases to make the process practical. This step is performed in two sub–steps: (1) First, a *quasi–exhaustive set of test cases* ($QuEST$) is computed by analyzing the rules and their input/output behavior. (2) Second, the large amount of test cases is limited by utilizing so–called validation criteria. Test cases that don't reach a certain validation necessity degree will be removed from $QuEST$ resulting in a *reasonably sized set of test cases* $ReST$. A workable compromise between these constraints is central to both the technique developed so far and the im-

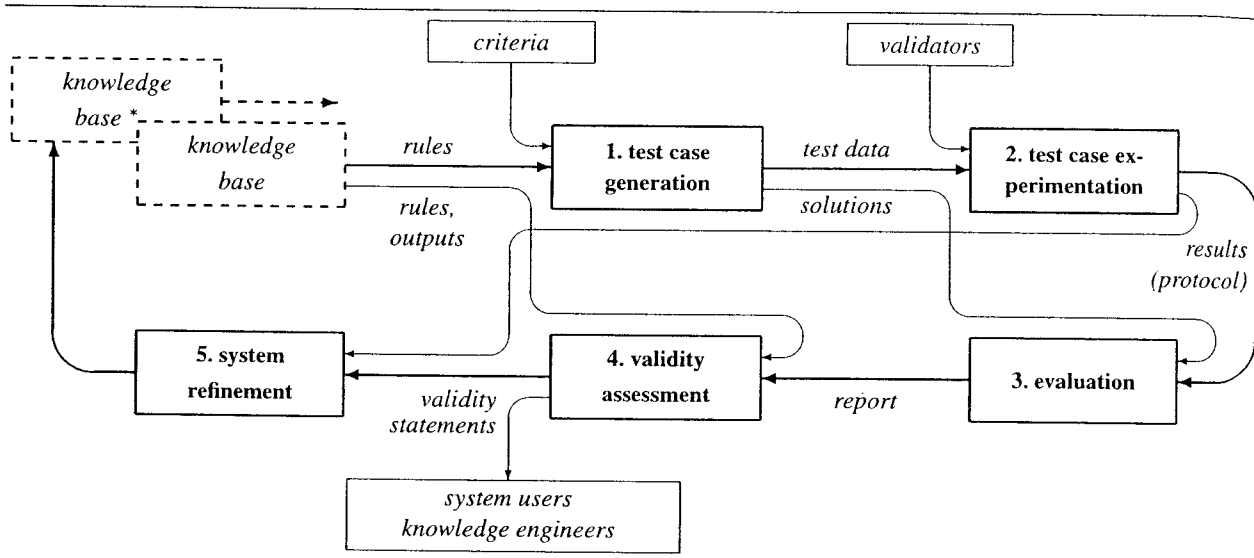

Figure 2. Steps in the Proposed Validation Process [12]

provements reached by introducing the VKB.

2. **Test case experimentation**: Intelligent systems emulate human expertise. Therefore, human opinion needs to be considered when evaluating the correctness of a system's response. Through a TURING Test – like validation approach, this step performs a fair evaluation of the correctness and/or dependability of a system's outputs given by imperfect human expertise. It consists of (1) exercising the set of test data by both the intelligent system and the validating experts and (2) presenting all results – those provided by the system as well as those provided by the human experts – to the validation panel anonymously.

3. **Evaluation**: The third step interprets the results of the experimentation and determines errors attributed to the system and reports it informally. As a side effect of the previous step, a test case competence assessment of the validators for each particular test case is computed and utilized for a more objective validity statement in the following step.

4. **Validity assessment**: In this step, the results of the evaluation are analyzed and conclusions about the system's validity are drawn. Depending on the purpose of the validation statement, the validity is expressed as (1) validity degrees associated to test cases, (2) validity degrees associated to the system's outputs, (3) validity degrees associated to system's rules, and finally (4) as a validity degree associated to the entire system.

5. **System refinement**: At the first view, the objective of validation is to gain reliable statements on the usefulness and dependability of an intelligent system. In the end, however, we are also interested in developing a more dependable system with a better performance. Therefore, this fifth step, which completes the framework, provides guidance on how to correct or decrease the effects of errors or vulnerabilities detected in the system as a result of the previous four steps. Since the validity assessment points out the rules which infer invalid solutions and the TURING Test experimentation reveals a so–called optimal solution to each test case, we are able to refine these rules with the objective to provide the optimal (i.e. most dependable) solution. This, naturally, leads to an improved input–output behavior of the system, and thus to a more dependable system.

The benefit of this standardized validation framework is that developers of knowledge–based systems can reference it when describing the validation process to the end user. This may enhance the acceptability of the system. Furthermore, this framework attempts to minimize the effort involved in validation of the expert system. This is because cases derived from the knowledge in the VKB don't have to be resolved in the process. The reason not to resolve them is that the VKB is intended to serve as a source of **external** knowledge, which consists of a historical solution that obtained good marks in the past. Lastly, this minimized effort leads to reduced and more predictable costs. A comprehensive description of all steps as well as the research behind this work can be found in [8]. Also [12] provides a more detailed description of this framework.

3. The Validation Knowledge Base Approach

In [17], a bi–directional, many–sided explanation typed multi–step validation method ($MMBV$) was proposed. Knowledge engineers (KEs) and computers can share validation knowledge with experts. By using this method, the workload on busy experts can be decreased significantly. For this purpose, the validation knowledge needs to be represented in computers. Therefore, the concept of a VKB and a validation approach based on it has been suggested (cf. [18], [19]). The basic idea is to reuse experts' validation experiences with the enjoyable effect of limiting the validation workload on busy experts.

However, there is a serious problem called the "knowledge acquisition bottleneck". It seems even more difficult to acquire validation knowledge than to acquire domain knowledge, because validation knowledge is a kind of meta–knowledge used for validating domain knowledge. TSURUTA et al. [19] suggest an approach, which is based on the concept that computers (supported by KEs and experts) acquire, validate, and refine validation knowledge in a VKB. Therefore, during the validation sessions all upcoming data is collected in a Validation Data Base (VDB). The VKB selects, pre–processes, and stores relevant historical data of VDB. Although validation (meta–) knowledge is difficult to acquire (also because experts are too busy to teach such validation expertise for various kinds of situations), some useful validation knowledge can easily be collected and incorporated as a VKB by analyzing validation sessions and memorizing their results. This way, their validation expertise can easily (and without the experts' support) applied to various kinds of situations.

Unfortunately, this knowledge is often different or inconsistent depending on the different expert opinions. The way to face this problem is explained in the following subsections. For implementation details see [17].

3.1. Experts' Validation Data Base: VDB

In the above–mentioned VKB approach, the validation knowledge is acquired through the data in the VDB of a validation system as introduced in [17]. Generally, the VDB is a protocol of test case evaluation procedures. Thus, it includes *test cases*, which consist of

- test data (test case inputs),
- test process data (test schedule and delay status, e.g.), and
- test results, i.e. (1) the test case solution, (2) explanations to this solution, (3) comments to this solution, as well as the data collected during the evaluation of this solution, i.e. (4) the evaluator and (5) the evaluation result itself (*valid* or *invalid*).

Based on these data, validation knowledge is automatically constructed and stored in the VKB as described below.

3.2. Validation Knowledge Base: VKB

As mentioned above, experts' validation data in the VDB includes test data (problems), solutions, and experts' validation results. They are considered to be experiences or examples of experts' validation knowledge. Now, these examples are acquired from the VDB and compiled to validation knowledge. This validation knowledge can be represented as (1) a case library [19] or (2) as a rule–base [18]. Either way, the VKB can be constructed from VDB by putting

- the test cases (problems with solutions) into (1) a case–condition part respectively (2) a rule's condition part, and
- the experts' validation results (expert's evaluation value with comments) into (1) a case–solution part respectively (2) rule's conclusion part.

For example, as to a Travelling Salesman Problem,

- the case–condition part (respectively the rule's condition part) is a problem (test data) such as a list of visited cities and constraints, accompanied with its solution such as an optimally ordered sequence of visited cities and
- the case–solution part (respectively the rule's conclusion part) is the expert's evaluation value such as OK (valid), NG (invalid) or a validity degree ranging from 1 to 5.

Each knowledge piece (either a case or a rule) of the VKB has various properties, such as a confidence value (CV), a many–sided explanation, an expert's comment, etc. Furthermore, in order to confirm the correctness of the acquired VKB, it has a property called $Supporter$, which is the list of experts who have accepted the knowledge piece, to trace back from where the validation knowledge originated (see [19]).

The validation and refinement of the acquired validation knowledge is necessary and important for a successful application of the introduced method. In the proposed approach, an acquired new validation knowledge piece (a new case or a new rule) is checked against the existing ones in the VKB. If an identical one is found, its confidence value (CV) is increased, and both are combined to one. However, if inconsistency exists, the CV is decreased [18], and the responsible experts are required to re–validate this knowledge piece by tracing back until the origin of the inconsistency is found. Other experts can be involved to assist if needed. This way each piece of validation knowledge is validated

and refined by the persons described in its $Supporter$ property indicating the persons responsible for the knowledge, namely the experts who made or accepted the validation results [19]. A wrong rule is removed or ignored under the control of CV or as a result of the above retrial.

Experts' validation knowledge can easily be acquired and incorporated as a correct and consistent VKB, since experts are usually too busy to teach or to validate such knowledge.

Thus, computers can automatically infer the validation results, utilizing the VKB and share the validation load of busy experts with the help of KEs who check and modify the automatic validation results. As a result, the validation load of busy experts is lightened.

4. Utilizing the Validation Knowledge base for the Turing Test

The objective of the approach is to utilize the experience made in a validation session for all upcoming ones. Therefore, we use a VKB to permanently store this historical experience.

In a first setting, the VKB needs to be involved in the steps *(1) test case generation* and *(2) test case experimentation* (see figure 2). This involvement is illustrated in figure 3. Furthermore, the data in the VKB is used in step *(3) evaluation*. Here, the data in VKB can be utilized to estimate the human experts' (historical) competence, which serves as a weight for a particular expert's rating of the solution, which is provided by the system under examination within the current validation session. Additionally, the knowledge in VKB can serve many other purposes. The incorporation of the VKB into the steps (1), (2), and (3) is described below. Chances and limits to use VKB within other steps of the framework are discussed in the introduction and outlook sections of this paper.

4.1. The Content of the VKB

Here we outline, which information needs to be stored and maintained in the VKB for the *test case experimentation*, in particular (1) the required input data, (2) the produced output data, and (3) additional necessary data. According to the formal settings in [8] and [13], the VKB contains a set of historical test cases, which can be expressed as 8–tuples $[t_j, sol_{K_j}^{opt}, E_K, E_I, r_{IjK}^*, c_{IjK}^*, \tau_S, D_C]$ with

- t_j being a test data (a test case input),
- $sol_{K_j}^{opt}$ being a solution associated to t_j,
- E_K being a list of experts who provided this particular solution,
- E_I being a list of experts who rated this solution,
- r_{IjK}^* being the historical rating of this solution, which is provided by the experts in E_I,
- c_{IjK}^* being the historical certainty[3] of this rating,
- τ_S being a time stamp associated with the validation session in which the rating was provided, and
- D_C being an informal description of the application domain C that might be helpful to explain similarities between different application domains or fields of knowledge.

Additionally, a *list of supporters* $E_S \subseteq E_I$ for each solution $sol_{K_j}^{opt}$ can be derived from this data. In particular, E_S is the list of rating experts, who provided a positive rating for $sol_{K_j}^{opt}$. For a comprehensive description of these data, see [13].

Of course, this database of historical knowledge is not completely transparent to all agents in the validation process. According to the purpose of the data in the VKB, some of it needs to be hidden. For example, to ensure the anonymity while solving and rating test cases within the TURING Test, E_K and E_I must not be presented to the expert panel of the current session. Furthermore, to ensure an unbiased rating, the historical rating r_{IjK}^* must not be presented to the expert panel that currently rates the solution.

4.2. Involvement of the VKB in the Test Case Experimentation

The intermediate results that occur during the experimentation as well as the VKB itself are stored in a relational database by using a *client–server database management system* ($DBMS$), which provides decentralized access to centralized data for clients who work independently from each other. The two *logical views* illustrated in figure 1 follow the same basic principle: All data is kept central to the view of *knowledge engineering (~server),* while only the necessary parts of it are shown to the *expert panel (~client)* (cf. [13]).

All experts of the panel independently take part in the TURING Test – like experimentation session. By utilizing an HTML–based implementation approach for the client application as developed in [13], each expert is almost free in the choice of time and place of his work. This effectively limits delays that are caused by experts who would otherwise be unavailable as well as the costs of the whole validation process.

As shown in figure 3, only those test cases in VKB, which "survived" the criteria–based reduction process, are

3 Besides providing a rating that might be 0 (wrong) or 1 (correct), the experts have the opportunity to express, whether (c=1) or not (c=0) they feel certain while providing this rating.

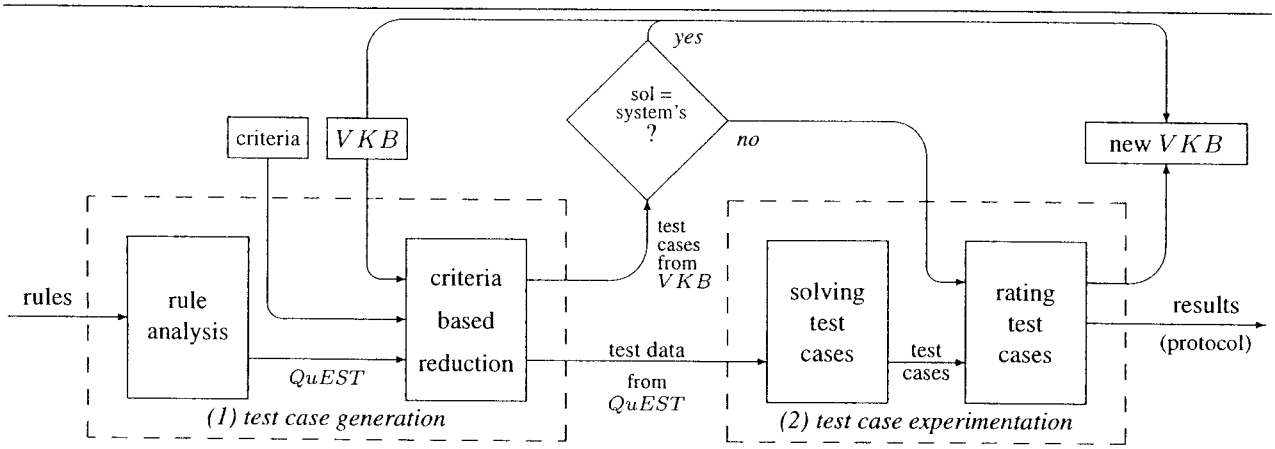

Figure 3. Incorporation of the Validation Knowledge Base (VKB)

used in the experimentation, because the criteria of the current application might differ from the ones of previous sessions. Since a VKB is a database of test cases and their associated solutions, which received an optimal rating in previous validation sessions, these solutions have to be seen as an additional (external) source of expertise that does not explicitly appear in the solving session (see figure 3).

Regardless of their former ratings, the cases from the VKB have to be rated by the expert panel. This has two basic reasons:

1. Topical domain knowledge of AI systems does have some dynamic characteristics, i.e. it might have changed since the time when the information in the VKB has been acquired. Reasons may be found in new topical insights, but also in application circumstances, that are different from the historical ones.
2. Additionally, there is a certain responsibility for the results of applying the validation framework, i.e. for the validity statements, which are developed, as well as for the refined knowledge base as a result of the entire cycle (see figure 2). These results need, when communicated and used for (commercial, political, ...) decisions, a clear association to responsible persons. Of course, the experts of the current panel which rated the solutions must serve as these responsible persons. Although there is already a (historical) rating for the test cases in the VKB, this panel must have the opportunity to provide their individual ratings to these test cases.[4]

Fortunately, not all cases of the VKB that "survived" the criteria–based reduction process need to be rated again: Only cases with solutions different from the systems solution have to be involved in the rating process, because (1) we are only interested in new external knowledge that is outside the expertise of the expert panel and (2) the systems solution is in the process anyway.[5]

4.3. Involvement of the VKB in other Steps of the Framework

Besides the incorporation of the VKB in the *test case experimentation* step as sketched in figure 3, the knowledge gained in the VKB is also applied for other useful purposes as newly proposed in the following, especially with a view at dependability such as breaking the limitation of human validators' dependability.

1. It can be used for a refined competence estimation of the experts in the panel. In the framework, this estimation is utilized as a weight of a certain expert's rating of the system's solution to compute its validity degree (cf. [8], [12]). Since all resulting validity statements are derived from these validity degrees, the refinement of the competence estimation leads to improved results of the entire framework. In fact, the consequence of better validity statements is a "more dependable" system after the refinement step (see figure 2). Furthermore, this competence estimation is very useful for selecting an appropriate expert panel.
2. Second, the VKB can support the identification of the optimal solution, which is the basis for the system refinement step (see figure 2) as well as for the updating process of the VKB itself (see next section below). In particular, if several solutions are candidates to be

4 Nobody would agree to be responsible for something that he/she can not control.

5 The test case generation step exclusively produces test cases with the system's solution. The test case solving session additionally provides alternative ("man–made") solutions to it.

the "optimal solution" (i.e. they receive the same approval by the expert panel), the information kept in the VKB is helpful to differentiate these candidates.

Both approaches are introduced in [13] and the basic algorithms are outlined below.

4.3.1. Competence Estimation of the Rating Experts Since the competence estimation of the experts is based on the experts' performance in the rating session, the ratings and certainties for the test cases originated from the VKB needs to be included in the estimation. The way to refine the approach in [8] accordingly is detailed in [13].

Since the VKB holds knowledge about the experts' competence in previous sessions, i.e. "historical competence", it opens the chance to select an appropriate expert panel for a scheduled session. Derived from the information in the VKB, [13] introduces

1. a *historical session competence* $sess\_est_{hist}(e_i, S'_i)$ of a certain expert e_i within a session S'_i,
2. a *historical competence trend* $trnd\_est_{hist}(e_i)$, which describes the development of an expert's competence over time,
3. an estimation of a *competence gain* $\Delta sess\_est_{hist}(e_i, \sigma_i^t)$ from one session to the next and an *average competence gain* $\delta_i(\sigma_i^t)$ over time,
4. a classification of experts as those with an (1) increasing, (2) even, and (3) decreasing competence over time, and
5. an *average historical competence* $avg\_est_{hist}(e_i)$.

Finally, [13] suggests a guideline to use the introduced concepts listed above for a qualified selection of an appropriate expert panel. Interestingly, the author itself claims to utilize these estimations with caution because they are based on data, which might be incomplete, irrelevant, and not representative. Furthermore, social reasons require handling all the concepts about an expert's competence with care and discretion.

4.3.2. Identification of the Optimal Solution For the 5th step of the framework, the *system refinement* (see figure 2), the concept of an optimal solution is introduced in [8]. This is, loosely speaking, the solution $sol_{opt}(t_j)$ to a test data t_j that gained the maximum approval by the experts in the current panel. Unfortunately, it might happen that there are several solutions, which enjoy the maximum approval. In these cases, the VKB is used to qualify one of these candidate solutions to be the "very best" one.

For this purpose, [13] introduces a step–by–step filtering process that is applied until one candidate solution is left over:

1. In a first step, the average competence of the experts, who are in the VKB in the *list of supporters* (see section 4.1) of the candidate solutions are considered. The candidate solution, which enjoys the maximal support by the VKB, is considered the "very best" one.
2. In case there are still several solutions for the step above, a *list of vetoers*[6] is derived from the VKB and their average competence is calculated from the data in the VKB. The candidate solution, which received the minimal "resistance" by the VKB, is considered the "very best" one.
3. If there are still several candidate solutions after the first two steps, the supporters for each of the remaining candidate solutions are compared: The solution that is supported by the expert e_i with the maximal competence $cpt(e_i, t_j)$ for the test data t_j, is considered the "very best" one.
4. The last opportunity to identify the "very best" solution, if there are still several ones after the three steps above, is a "run–off" session with the expert panel and the remaining candidate solutions.

4.4. Maintenance of the VKB

To ensure that the VKB really gains experience while being used, it has to be updated within each validation session. Updating, in this context, means adding new cases to the VKB. Of course, the 8–tuples introduced in section 4.1 are not stored as physically different entries, because an optimal storage and access is managed by the *client–server database management system* ($DBMS$).

The VKB stores the historical cases explicitly and associated to the right (historic) context by marking it with a *time stamp*. Thus, it eliminates the opportunity for misinterpretations. Since historical knowledge from the VKB is always *revalidated* within the current session, *invalid facts* are sorted out by utilizing the *meta–knowledge* [7] of the human experts.

In fact, the *experience* of a session, which is worth keeping, is the optimal ("very best") solution sol_{Kj}^{opt} to each test data t_j that has been solved within the session (cf. section 4.1). Additionally, the associated list of solvers E_K and the list of raters E_I needs to be kept in the VKB. Furthermore, a *time stamp* has to be provided for each new case in the VKB. The *time stamp* τ_S of the current experimentation session is assumed to be the starting time of the *rating session*. In fact, the only requirement the time stamps have to meet is that they have to be determined

6 Vetoers are experts, who provided a negative rating for a considered solution.

7 Meta–knowledge is "knowledge about knowledge", i.e. about its retrieval, context, usage, etc.

in the same way in each and every session to maintain their *order over time.*[8] By adding a description of the *application domain* and *context* D_C, all resulting 8–tuples $[t_j, E_K, E_I, sol^{opt}_{Kj}, r^*_{IjK}, c^*_{IjK}, \tau_S, D_C]$ have to be stored in the *VKB*.

4.5. The Validation Expert Software Agent (VESA)

With the view at dependability, the VKB itself is also extended. Namely, the concept of $VESA$ is proposed to further break the limitation of a human validator's dependability through storing personal historical validation knowledge, namely previous validation judgments or experiences of each human expert. A $VESA$ obtains and stores validation knowledge / data autonomously from validation results of the experts (validators) participating in the TURING Test, namely test case experimentation protocol. In this meaning, the concept of $VESA$ is considered as the extended VKB. However, each $VESA$ is basically an autonomous software agent corresponding to each human expert validator. Each $VESA$ gains personal validation knowledge mainly from personal data such as (not always best) solutions, ratings, etc. of the human expert validator corresponding to it. On the other hand, the original VKB gains knowledge from data concerning the best rated solution. Thus, each $VESA$ is an intelligent agent corresponding to each human validation expert. In every validation session, they become more intelligent as well as more adaptive to wider (similar but slightly different) applications, since they can learn from test inputs, the associated answers, their certainties and their ratings provided by the human validators. Namely, they increase their validation competence through validation knowledge gained by various sessions over time.

Though a $VESA$ is an agent of a human validation expert, it can also gain the validation knowledge / data of other validators, e.g. test data, the solution and its rating when a very high–rated (but not always best) solution happens to be derived by one of the same type of validators which have usually almost the same solutions with each other. Further, it can be an agent representing a group or an organization of validation experts. Thus, it can become more and more competent. Since they are not human but machine, anonymity will be kept even if they get information of other (human) experts. They do not need the name of each expert, but instead of the name, they need an ID only to distinguish, whether or not the information belongs to the same expert. This concept of $VESA$ contributes to dependable validation which leads to dependable AI systems, as follows:

1. $VESA$ can replace the human expert when he is too busy to participate in validation.
2. $VESA$ can be a competent validator and upgrade test case experimentation and test case generation.
3. A group of $VESA$s might do test case experimentation without experts, since each $VESA$ has different validation knowledge and can be tested from various views.

Therefore, the $VESA$ concept contributes to dependability of AI systems, though many AI systems do not have a commonly accepted knowledge standard.

5. Summary and Conclusion

With the view of increasing system dependability, this paper presented a synergistic combination of the VKB approach and the TURING Test – like validation approach, which makes AI system validation and thus the AI system itself more dependable.

The historical validation knowledge in a VKB can be used to keep an ever–improving benchmark for periodic validation of an intelligent system.

This involvement of a VKB led to several significant advantages as follows:

- Firstly, it lightens the burden on the human experts who are called upon to serve as validators. Such individuals are typically very busy, not to mention expensive.
- Secondly, it enables the improvement of the validation methodology itself, e.g. (1) by using the validation knowledge to select appropriate experts for the validation panel as well as (2) by using it to refine the concept of the competence estimation of the involved experts.
- Thirdly, it provides a mechanism to continually update / upgrade the test case set to reflect the latest findings about the domain, and to identify an optimal solution among several solutions each of which has equal or almost equal ratings.

Especially the latter (second and third) advantages lead to better validation results, i.e. more dependable validity statements, due to better estimations of the validity degrees of the executed test cases. As a result, these advantages make AI / intelligent systems more dependable, since such dependability is based on dependable validity statements.

To demonstrate the usability of the approach, a prototype application TestMeToo (**Test** case experi**M**entation **Too**l) has been developed by KURBAD (cf. [13]).

Lastly, as an extension to the VKB, the concept of the $VESA$ is introduced. Each $VESA$ gains personal validation knowledge automatically from validation results of its

8 This is important for the estimation of the *historical competence trend* $trnd\_est_{hist}(e_i)$ of an expert e_i as detailed in subsection 4.3.1.

corresponding human expert, while the original VKB gains representative validation knowledge from data concerning the best rated of all solutions. Therefore, each $VESA$ is an intelligent agent corresponding to each human validation expert. Through breaking limitations of human validator's dependability, $VESA$ aims at improving the dependability of AI systems. It suggests a way to systematically "construct" human–like validators by learning their solving and rating behavior. This brings a new dimension for AI system validation and its dependability, though there is much limitation or much to be researched.

Acknowledgements

The authors are grateful to any individuals who supported the work behind this paper. In particular, we gratefully acknowledge Professor Dr. Yoshihiro Tohma, the president of the Tokyo Denki University, for his excellent leadership in our research as well as in the university, and for his fruitful topical contributions to the previous conferences such as Keynote speech at PRDC–2002. We also acknowledge Mr. Keiichi Uehara in the Tokyo Denki University for our fruitful discussions with him. Furthermore, we acknowledge Mr. Hideo Ooba (National Space Development Agency of Japan) for the generous support of our research.

References

[1] Adrion, W.; Branstad, M.; Cherniavski, J. 1982. Validation, verification and testing of computer software. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21, # 2, pp. 293–301.

[2] Barr, V. 1996. Applications of rule–base coverage measures to expert system evaluation. Rutgers University, Technical Report DCS–TR–340.

[3] Ginsberg, A.; Weiss, S.; Politakis, P. 1985. SEEK2: A generalized approach to automatic knowledge base refinement. *Proc. of the 9th International Joint Conference on Artificial Intelligence 1985 (IJCAI–85)* Los Angeles, CA, USA, Morgan Kaufmann, pp. 367–374.

[4] Ginsberg, A. 1988. Knowledge–base reduction: a new approach to checking knowledge bases for inconsistency and redundancy. *Proc. of the 7th Annual National Conference on Artificial Intelligence 1988 (AAAI–88)* St. Paul, MN, USA, AAAI Press / The MIT Press, pp. 367–374.

[5] Glenford, J.; Myers, S. 1976. *Software Riliability – Principles and Practices*. New York: John Wiley and Sons.

[6] Gonzalez, A.J.; Barr, V. 2000. Validation and verification of intelligent systems – what are they and how are they different? *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 12, # 4, pp. 407–420.

[7] IEEE, 1990. IEEE Std. 610.12–1990, Glossary of Software Engineering Terminology.

[8] Knauf, R. 2000. *Validating Rule–Based Systems – A Complete Methodology*. Habilitation Thesis, Ilmenau Technical University, Faculty of Computer Science and Automation, ISBN 3-8265-8293-4 Aachen: Shaker.

[9] Knauf, R.; Gonzalez Avelino J.; Tsuruta, S. 2003. Utilizing validation experience for system validation. I. Russell and S. Haller (Eds.): *Proceedings of the* 16^{th} *International Florida Artificial Intelligence Research Society Conference 2003 (FLAIRS 2003)*, St. Augustine, FL, USA, Menlo Park, California: AAAI Press, pp. 223–227.

[10] Knauf, R.; Philippow, I.; Gonzalez, A.J. 2000. Towards validation and refinement of rule–based systems. *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 12, # 4, pp. 421–431.

[11] Knauf, R.; Philippow,I.; Gonzalez, A.J.; Jantke, K.P.; Salecker, D. 2002. System refinement in practice – using a formal method to modify real life knowledge. Kohlen (ed.): *Proc. of the* 15^{th} *International Florida Artificial Intelligence Research Society Conference 2002 (FLAIRS-02)*, Pensacola Beach, FL, USA, Menlo Park, CA: AAAI Press, pp. 216–220.

[12] Knauf, R.; Gonzalez, A.J.; Abel, T. 2002. A framework for validation of rule–based systems. *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, vol. 32, # 3, pp. 281–295.

[13] Kurbad, T. 2003. *A Concept to Apply a Turing Test Technology for System Validation that Utilizes External Validation Knowledge*. Diploma Thesis, Technical University Ilmenau, School of Computer Science and Automation, Library index # 2003-09-03/053/IN95/2238, 2003.

[14] O'Keefe, R.M.; O'Leary, D:E: 1993. Expert system verification and validation: a servey and tutorial. *Artificial Intelligence Review*, vol. 7, pp. 3–42.

[15] Preece, A.D.; Grossner, C.; Chander, P.G.; Radhakrishnan, T. 1993. Structural validation of expert systems using a formal method. *Proc. of the 11th Annual National Conference on Artificial Intelligence 1993 (AAAI–93), Workshop on Validation and Verification of Knowledge–Based systems*, Washington D.C., USA, AAAI Press / The MIT Press, pp. 19–26.

[16] Smith, S. 1991. *Verification and validation of rule–based expert systems*. Doctoral Dissertation, Florida State University.

[17] Tsuruta, S.; Onoyama, T.; Kubota, S; Oyanagi, K. 2000a. Validation method for intelligent systems. Etheredge / Manaris (eds.): *Proc. of the* 13^{th} *International Florida Artificial Intelligence Research Society Conference (FLAIRS-00)*, Orlando, FL, USA, pp. 361–365.

[18] Tsuruta, S.; Onoyama, T.; Kubota, S; Oyanagi, K. 2000b. Knowledge-based approach for validating intelligent systems. Kern (ed.): *Proc. of* 45^{th} *Internat. Scientific Colloquium (IWK-00)*, Ilmenau, Germany, Technical Univ. of Ilmenau, pp. 769–774.

[19] Tsuruta, S.; Onoyama, T.; Taniguchi, Y. 2002. Knowledge–based validation method for validating intelligent systems. Kohlen (ed.): *Proc. of the* 15^{th} *Internat. Florida Artificial Intelligence Research Society Conference 2002 (FLAIRS-02)*, Pensacola Beach, FL, USA, Menlo Park, CA:AAAI Press, pp. 226–230.

[20] Turing, A.M. 1950. Computing machinery and intelligence. *Mind*, LIX, number 236, pp. 433–460.